MACHINE LEARNING MODELS ON AIR QUALITY DATASET

Kakarla Nanda Vennela
Raghava Ponugoti
Bhavishya Chowdary Katragadda

December 12, 2022

1 Introduction

Around the world, nine out of ten people breathe unhealthy air. Air pollution has become one of the most serious challenges globally and it affects everything on this planet. It is harmful to the health of living beings and it also impacts the environment which results in various consequences such as climate change and global warming. Therefore, many countries in the world built air pollution monitoring and control stations in many cities to observe the levels of various air pollutants like CO, NO2, SO2 and to alert the citizens about air pollution index which exceed the quality threshold.Some of these harmful air pollutants such as Nitrogen Dioxide primarily get into the air through combustion of fuels. Also, vehicular emission contributes to the majority of Carbon Monoxide let into our atmosphere. The presence of poisonous gases and substances in the air causes air pollution; hence it is impacted by the meteorological factors of a particular place such as temperature and humidity. This project provides support in order to frame air quality standards and regulations based on the analysis performed on various gas concentrations causing air pollution and also based on the meteorological data of the place.

2 Data

The air quality dataset for our project is collected from the UCI Repository and this dataset is available in CSV format. This dataset contains data of the average hourly responses of various elements in air for nearly a year, ie. from March 2018 to April 2019. The dataset consists of 9357 rows and 15 columns. The attributes present in the dataset are displayed in the table below.

S.no	Attribute Description
0	Date in (DD/MM/YYYY) format
1	Time in (HH.MM.SS) format
2	CO concentration in mg/mˆ3
3	PT08.S1 (Tin Oxide, CO targeted) concentration
4	Non Metanic Hydro Carbons concentration in microg/mˆ3
5	Benzene concentration in microg/mˆ3
6	PT08.S2 (Titania, NMHC targeted) sensor response
7	NOx concentration in ppb
8	PT08.S3 (Tungsten Oxide, NOx targeted) concentration
9	NO2 concentration in microg/mˆ3
10	PT08.S4 (Tungsten Oxide, NO2 targeted) concentration
11	PT08.S5 (Indium Oxide, O3 targeted) concentration
12	Temperature in Aˆ°C
13	Relative Humidity
14	Absolute Humidity

Also, the air quality standards for these attributes in unpolluted air are displayed in the table below.

Attribute	Standard range in air
CO	0.06 to 0.14 mg/m3
NO2	150 to 2055 mg/m3
Ozone	120 mg/m3
Benzene	975 to 9750 mg/m3
Titanium Oxide	2.4 mg/m3
Tungsten Oxide	0.14 to 6.8 mg/m3
Tin Oxide	P0.072 to 5.4 mg/m3
Indium Oxide	0.018 to 9.8 and 0.072 to 5.4 mg/m3

3 Data Prepossessing

Data Preprocessing is a technique that involves transforming the raw data into an understandable format. In this technique, the data is cleansed through processes such as filling in missing values. As it contains some missing value, the dataset is cleaned, and decimal values are converted into proper float values. The dataset is downloaded and imported to the project by mentioning the Google Drive location of the downloaded dataset and mounting drive in Google Colaboratory software.

```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
import seaborn as sns
import warnings

warnings.filterwarnings("ignore")

file_path ="/content/drive/MyDrive/AMAT_502_Project/AirQualityUCI.csv" #To read data from CSV file
data_set = pd.read_csv(file_path)
```

Figure 1: Loading Dataset in Google Colaboratory.

4 Splitting training and testing dataset

Separating dataset into training and testing datasets is an important part of evaluating data mining models. Typically, while separating a data set into a training dataset and testing dataset, most of the data is utilized for the training process, and a smaller portion of the data is utilized for testing. After a model has been developed using this training dataset, test the model is by making predictions against the test dataset. Using the same data for the training and testing process will minimize the discrepancies effects of data and helps in a better understanding of the characteristics of the model.

5 Correlation

Correlation is a statistical measure that expresses the extent to which two variables are linearly related(i.e they change together at a constant rate). It is a common tool for describing simple relationships without making a statement about the cause and effect. Correlation among numerical values from our dataset is shown in figure 2 by using heatmap.

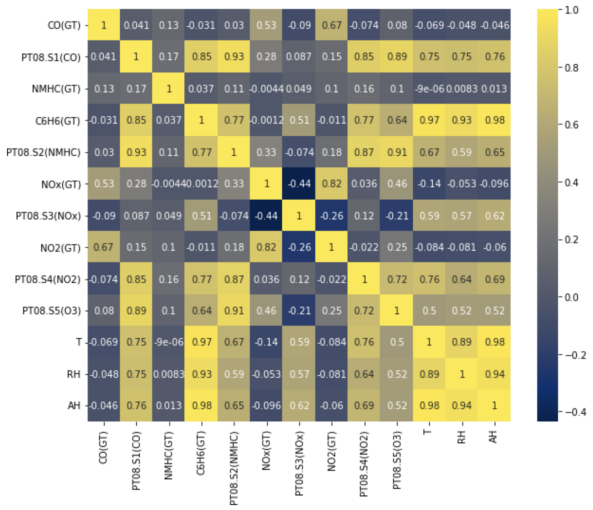

Figure 2: Correlation among numerical values.

6 Classification

6.1 What is Classification?

Classification is a supervised Machine Learning Algorithm. It is the process of categorizing the given

set of data into classes .Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). Here the targets are also provided with the input data.

6.2 Random Forest Classifier

6.2.1 What is a Random Forest Classifier?

The Random Forest Classifier is a meta-estimator that fits a forest of decision trees . It uses averages to improve prediction accuracy. It also aggregates the votes from different decision trees to decide the final class of the test object. This classifier is one of the most-used algorithms, due to its simplicity and diversity and performs well in real time datasets. Random Forest Classifiers are more accurate and flexible. This approach to classification is similar to the decision tree, except the questions that are posed include some randomness. The goal is to push out bias and group outcomes based on the most likely positive responses.

It is an ensemble tree-based learning algorithm. The Random Forest Classifier is a set of decision trees from a randomly selected subset of the training set. It aggregates the votes from different decision trees to decide the final class of the test object.

Random Forest Prediction for classification problem
f(x)=majority vote of all predicted class over N tree

The model was trained with our dataset by dividing the dataset into a 70:30 ratio where 70% of the data was used to train the model and the rest 30% was used to test the trained model.

6.3 Result

The classification report helped us in understanding the overall performance of our trained model. It shows the main classification metrics such as precision, recall, and f1-score on a per-class basis. The metrics are calculated by using true and false positives, and true and false negatives. Positive and negative in this case are generic names for the predicted classes. There are four ways to check if the predictions are right or wrong.

TN / True Negative: when a case was negative and predicted negative

TP / True Positive: when a case was positive and predicted positive

FN / False Negative: when a case was positive but predicted negative

FP / False Positive: when a case was negative but predicted positive

Precision is the ability of a classifier not to label an instance positive that is negative. For each class, it is defined as the ratio of true positives to the sum of true and false positives.

TP – True Positives
FP – False Positives

Precision: Accuracy of positive predictions.

$$Precision = TP/(TP + FP)$$

Recall is the ability of a classifier to find all positive instances. For each class, it is defined as the ratio of true positives to the sum of true positives and false negatives.

Recall: Fraction of positives that were correctly identified.

$$Recall = TP/(TP + FN)$$

	precision	recall	f1-score	support
-200	1.00	1.00	1.00	121
0	0.96	0.96	0.96	1329
1	0.95	0.96	0.96	1338
2	1.00	0.15	0.26	20
accuracy			0.96	2808
macro avg	0.98	0.77	0.79	2808
weighted avg	0.96	0.96	0.96	2808

Figure 3: Classification Report.

From Figure 4 we can observe that we obtained Temperature as the important feature.

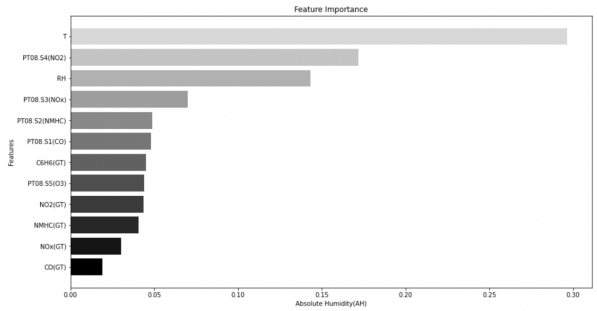

Figure 4: Visualizing feature importance.

7 Regression

7.1 What is Regression?

Regression is basically a supervised Machine Learning method, i.e. It is a statistical method to model the relationship between a dependent(target) and independent(predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. There are several types of regression models. They are:

1. Linear Regression,
2. Logistic Regression,
3. Random Forest Regression,
4. Polynomial Regression,
5. Stepwise Regression,
6. Ridge Regression,
7. Lasso Regression,
8. ElasticNet Regression,

Among the above models, we used Linear Regression and Random Forest Regression models in this project.

7.2 Regression analysis

The processed data sets are used to create a function to plot the training and validation data for the different models such as Linear Regression, Support Vector Regression, Decision Tree Regression, and Lasso Regression. Linear Regression is a basic and best-used type of predictive analysis. Linear Regression is used to examine two things; namely, it checks whether a set of predictor variables is doing a good job in predicting an outcome (dependent) variable and checking which variables, in particular, are the significant predictors of the outcome variable.

7.3 Air quality prediction

Air quality is predicted using the Root Mean Square Error(RMSE), and it tells how well a regression model can predict the values of a response variable in absolute terms. Root Mean Square Error is the standard deviation of the residuals(prediction errors). Residuals provide a measure of the distance of the data points from the Regression line. Root Mean Square Error is a measure of how spread out these residuals are, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results. The lower the Root mean square error, the better a given model can fit a dataset. However, the range of the dataset we are working with is important in determining whether a given Root mean square error value is low or not.

7.4 Linear Regression

7.4.1 What is Linear Regression?

Linear Regression analysis is used to predict the value of a target variable based on the values of other variables or independent variables by fitting a linear equation between the dependent and independent variables. It employs a regression line, also known as a best-fit line. The linear connection is defined by the equation $y = mx + c$, where m is the slope of the regression line and c is the constant and y is the intercept.

7.4.2 Linear Regression Results

We calculated the Linear Regression score. It tells us how well our model is fitted to the data by comparing it to the average line of the dependent variable. Greater the percentage of the score, the greater the performance of our model.

```
print('Testing Accuracy:',lr.score(X_test, Y_test)*100)  #Score between x_test and y_test
Testing Accuracy: 99.96883240890604
print('Training Accuracy:',lr.score(X_train, Y_train)*100) #Score between x_train and y_train
Training Accuracy: 99.97385551860269
```

Figure 5: Testing Accuracy and Training Accuracy

From Figure 5, we can observe that we obtained a testing accuracy of 99.96883240890604% and a training accuracy of 99.97385551860269%. That means we are getting around 0.0312% error and 0.0261% error respectively. So, we calculated the mean square errors by importing `mean_absolute_error`, `mean_square_error` from `sklearn.metrics`.

Mean Absolute Error(MAE), Mean Square Error(MSE), and Root Mean Square Error(RMSE) are calculated by using the formulae:

$$\text{Mean Absolute Error} = \frac{\sum_i^n (y_t - y_p)}{n}$$

$$\text{Mean Square Error} = \frac{\sum_i^n (y_t - y_p)^2}{n}$$

$$\text{Root Mean Square Error} = \sqrt{\frac{\sum_i^n (y_t - y_p)^2}{n}}$$

where y_t = testing value
y_p = predicted value
n = number of samples

The mean absolute error, mean square error, root mean square error of the predicted model are shown figure 6.

```
mean_sq=np.sqrt(mean_squared_error(Y_test,Y_pred))
print('Root Mean Squared Error:',mean_sq)
Root Mean Squared Error: 0.22532948202734948
print('Mean Squared Error:',mean_squared_error(Y_test, Y_pred))
Mean Squared Error: 0.050773375470713616
print('Mean Absolute Error:',mean_absolute_error(Y_test, Y_pred))
Mean Absolute Error: 0.18181574423236183
```

Figure 6: Errors

7.4.3 Plot between observed and predicted:

Figure 7 shows the plot between observed value and predicted value i.e., y test and y pred respectively. As we obtained an accuracy of 99.96883240890604% we can say that most of the predicted values matched with the observed values.

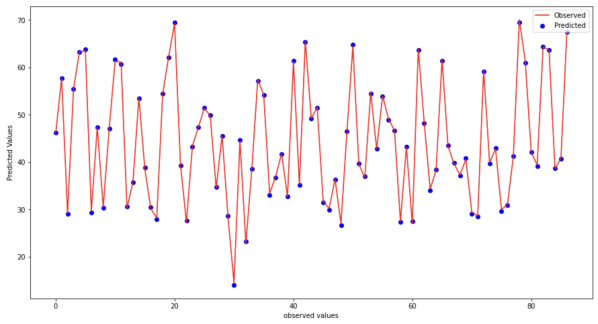

Figure 7: Observed vs Predicted Plot

The only way we can evaluate our regression model is by comparing its performance with other regression models. So, we compared our Linear Regression model with Random Forest Regression model.

7.4.4 Linear Regression Plot:

As shown in Figure 8, a plot is drawn between observed No2 Concentrations and expected Relative Humidity and a regression line is drawn for this plot and it is shown in the same plot.

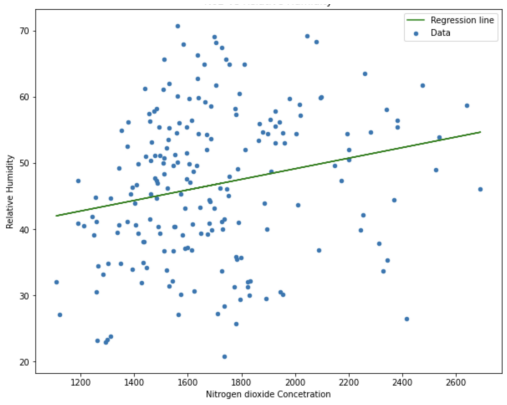

Figure 8: Linear Regression model

7.5 Random Forest Regression

7.5.1 What is Random Forest Regression?

Random Forest Regression is a supervised Machine Learning Algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. In other terms, it constructs several decision trees during training period and predicts the result by taking sum of all sub-tree predictions divided over total number of trees(N).

7.5.2 Comparing Linear Regression and Random Forest Regression

We have calculated the outputs to compare both Linear Regression and Random Forest Regression models. For easy comparison, we have created a dataframe with the parameter's mean absolute error, mean square error, and root mean square error.

	Model	Mean Squared Error	Root Mean Squared Error	Mean Absolute Error
0	Linear Regression	0.050773	0.225329	0.181816
1	Random Forest Regression	1.390828	1.179334	0.694862

Figure 9: Comparison of Linear Regression and Random Forest Regression

From Figure 9, we can observe that the errors Mean Absolute Error, Mean Square Error, Rort Mean Square Error of the Linear Regression model are lower than the Random Forest Regression model. Hence, we can conclude that Linear Regression gave the better prediction.

8 Clustering

8.1 What is Clustering?

Clustering is basically an unsupervised Machine Learning method, i.e. a method in which we draw references from datasets consisting of input data without any labeled responses. Clustering deals with finding a structure in a collection of unlabelled data. We deal with Clustering in almost every aspect of our daily lives. It is the subject of active research in various fields such as Statistics, Pattern Recognition and Machine Learning. Clustering methods can be classified into various types such as Hierarchical,Flat, Partitioning and Grid Based, etc. In our project, we have used K-Means clustering.

8.2 K-Means Algorithm

8.3 What is K-Means Algorithm?

K-Means Algorithm is one of the simplest unsupervised Machine Learning algorithms for solving clustering problems in Machine Learning or Data Science. When the output or response variable is not provided, this algorithm is used to categorize the data into distinct clusters. It is an iterative algorithm which divides the unlabeled dataset into k different clusters in such a way, that the data point in each cluster has similar properties. Being a centroid based algorithm, where each cluster is associated with a centroid point, the main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. It is also known as a data-driven Machine Learning approach since it clusters data based on hidden patterns, insights and similarities in the data.

8.3.1 How to Use K-Means Algorithm?

- Select the number of clusters(denoted as K) for the dataset.
- Select K number of centroids.
- By calculating the Euclidean distance or Manhattan distance, assign the points to the nearest centroid, thus creating K groups.
- Now find the original centroid in each group.
- Again, reassign the whole data point based on this new centroid, then repeat the above step until the position of the centroid doesn't change.

8.3.2 Elbow Method

In the K-Means Clustering algorithm, 'K' value defines the number of predefined clusters that needed to be created in the process. Also, the performance of the K-Means clustering algorithm depends upon highly efficient clusters that it forms. Hence, in cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a dataset. Here, we are varying the number of clusters from 1-10. For each value of K, we calculate the Within Cluster Sum of Squares(WCSS) value. For example, let us assume our k=3, then we calculate Within Cluster Sum of Squares value as follows:

$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Within Cluster Sum of Squares is the sum of the squared distance between each point and centroid in a cluster. After calculating the WCSS values, a graph is plotted with K value on X-axis and WCSS value on Y-axis. As the number of clusters increase, the WCSS value will start to decrease. WCSS value is largest when k=1. When we analyze the graph, we can observe that the graph rapidly changes at a point, thus creating an elbow shape. From this point, the graph moves almost parallel to X-axis. This K value point at which a sharp bend occurs is said to be the best K value for this given dataset for efficient results.

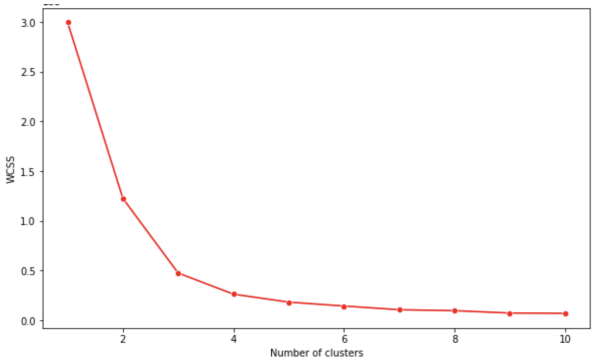

Figure 10: Elbow Method Plot.

8.4 Result

As shown in Figure 10, we have obtained k=3 as the perfect value for implementing the K-Means algorithm. Hence, in figure 11, there are 3 clusters in total which are visualized in different colors and the centroid of each cluster is visualized using a red star. The parameters chosen for this clustering and visualization are CO(GT) AND PT08.S3(NOx)

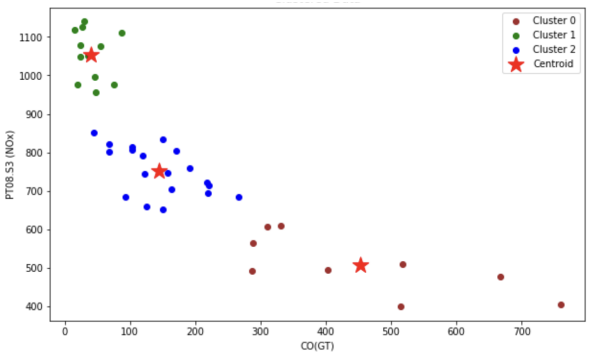

Figure 11: K-Means Visualization

9 Conclusion

Prediction of air quality is a challenging task because of the dynamic environment, unpredictability, and variability in space and time of pollutants. The grave consequences of air pollution on humans, animals, plants, monuments, climate, and environment call for consistent air quality monitoring and analysis, especially in developing countries. The short term exposure to various gases present in air causes headaches, dizziness, vomiting, nausea, coughing and difficulty in breathing while the long term exposure causes irregular heartbeat, non-fatal heart attack and even death due to lung disease. Hence, the main sources of these gases are such as motor vehicles, power stations, and cigarette smoking should be minimized. Hence, this project is useful to know about the con- centration of gases in the air. It creates awareness among people about the air quality standards and regulations based on the issues of toxic and pathogenic air exposure and health-related hazards for human welfare.In our project, the dataset is split into train-test subsets by the ratio of 70–30% respectively. The results of Machine Learning models for both the train-test subsets are presented in terms of standard metrics like accuracy, precision, recall, and F1-Score. Then, the classical statistical error metrics, namely Mean Absolute Error, Root Mean Square Error, and Mean Square Error are evaluated to assess and compare the performances of Machine Learning models.

10 Contributions:

Classification - Nanda Vennela Kakarla
Regression - Raghava Ponugoti
Clustering - Bhavishya Chowdary Katragadda